What is claimed is:

Sub A47

1. A system for program language processing for translating source programs to generate an object program, comprising:

a preprocessor for executing preprocessing of source programs inputted in translation units;

a data type definition table, arranged for one object program, for registering a data type definition for data or a function in the source programs;

a code optimizing processor for scanning all the preprocessed source programs to be used as a source for generating the object program, and deleting a duplicate data type definition from the source programs to optimize the source programs;

a language processor for compiling the optimized source programs; and

a software driver for controlling a transfer of a source program and a processing result of at least one of the preprocessor, the code optimizing processor, and the language processor.

2. The system according to claim 1, wherein the code optimizing processor includes:

a data type definition detection unit for detecting a predetermined data type definition from the preprocessed source program;

a first decision unit for deciding whether definition information of the detected data type definition is registered into the data type definition table or not;

a first registration unit for registering the definition information of the data type definition into the data type definition table when it is decided by the first decision unit that the definition information of the data type definition does not have been registered; and

Sub A16

a first deletion unit for deleting the data type definition detected by the data type definition detection unit from the preprocessed source program when it is decided by the first decision unit that the definition information of the data type definition has been registered.

3. The system according to claim 2, wherein the code optimizing processor further includes:

an instantiation request detection unit for detecting an instantiation request of a data type definition from the preprocessed source program;

a second decision unit for deciding, with reference to instantiation information in the data type definition table, whether the instance of a data type definition corresponding to the detected instantiation request of the data type definition has been generated or not; and

an instance generation unit for generating the instance of the data type definition when it is decided by the second decision unit that the instance of the data type definition does not have been generated, for registering information representing the generation of the instance into the data type definition table as the instantiation information, and

for suppressing generation of the instance of the data type definition when it is decided by the second decision unit that the instance of the data type definition has been generated.

4. The system according to claim 2, wherein the code optimizing processor further includes:

a second deletion unit for deciding the presence/absence of usage of data type in the data type definition table and deleting a definition for a data type which is not used in all the source programs to be used as a source for generating

Sub A16

the object program from the source programs.

5. The system according to claim 2, wherein the data type is one of a multiphase type data, a multiphase type function, and a multiphase type holding member function.

6. The system according to claim 5, wherein the data type definition table includes at least instantiation information representing whether instantiation is requested in the source program for every symbol of each data type of the multiphase type.

7. The system according to claim 6, wherein the data type definition table includes member usage information representing, when the data type is a multiphase type holding member function, whether each member function is used or not, and

the optimizing processor determines member function of a multiphase type the instance of which is to be actually generated in the source program with reference to the member usage information in the data type definition table.

8. The system according to claim 3, wherein the instance generation unit of the code optimizing processor converts the name of the data definition into an unique name in one source program.

9. A method of program language processing for translating source programs to generate an object program, comprising the steps of:

executing preprocessing of source program inputted in translation units;

scanning all the preprocessed source programs to be used

Sub A16

as a source for generating the object program;

deleting a duplicate data type definition from the source program with reference to a data type definition table arranged for one object program, the table registering a data type definition for data or a function in the source program to optimize the source program; and

compiling the optimized source program.

10. The method according to claim 9, wherein the optimizing step includes the steps of:

detecting a predetermined data type definition from the preprocessed source program;

deciding whether definition information of the detected data type definition is registered into the data type definition table or not;

registering the definition information of the data type definition into the data type definition table when it is decided that the definition information of the data type definition does not have been registered; and

deleting the data type definition detected by the data type definition detection unit from the preprocessed source program when it is decided that the definition information of the data type definition has been registered.

11. The method according to claim 10, wherein the optimizing step further includes the steps of:

detecting an instantiation request of a data type definition from the preprocessed source program;

deciding, with reference to instantiation information in the data type definition table, whether the instance of a data type definition corresponding to the detected instantiation request of the data type definition has been

sub A17

generated or not; and

generating the instance of the data type definition when it is decided that the instance of the data type definition does not have been generated, registering information representing the generation of the instance into the data type definition table as the instantiation information, and

suppressing generation of the instance of the data type definition when it is decided that the instance of the data type definition has been generated.

12. The method according to claim 10, wherein the optimizing step further includes the step of:

deciding the presence/absence of usage of data type in the data type definition table and deleting a definition for a data type which is not used in all the source programs to be used as a source for generating the object program from the source programs.

13. The method according to claim 10, wherein the data type is one of a multiphase type data, a multiphase type function, and a multiphase type holding member function.

14. The method according to claim 13, wherein the data type definition table includes at least instantiation information representing whether instantiation is requested in the source program for every symbol of each data type of the multiphase type.

15. The method according to claim 14, wherein the data type definition table includes member usage information representing, when the data type is a multiphase type holding member function, whether each member function is used or not, and

Sub A16

the optimizing step determines member functions of a multiphase type the instance of which must be actually generated in the source program with reference to the member usage information in the data type definition table.

16. A computer readable recording medium for causing a computer to execute program language processing for translating source program to generate an object program, comprising:

a process for executing preprocessing of source program input in translation units;

a process for scanning all the preprocessed source programs to be used as a source for generating the object program;

a process for deleting a duplicate data type definition from the source programs with reference to a data type definition table arranged for one object program so as to suppress instantiation of a data type definition which has been instantiated as needed to optimize the source program, the table registering a data type definition for data or a function in the source program; and

a process for compiling the optimized source program.

17. The medium according to claim 16, wherein the optimizing process includes:

a process for detecting a predetermined data type definition from the preprocessed source programs;

a process for deciding whether definition information of the detected data type definition is registered into the data type definition table or not;

a process for registering the definition information of the data type definition into the data type definition table when it is decided that the definition information of the data type definition does not have been registered; and

Sub A16

a process for deleting the data type definition detected by the data type definition detection process from the preprocessed source program when it is decided that the definition information of the data type definition has been registered.

18. The medium according to claim 17, wherein the optimizing process further includes:

a process for detecting an instantiation request of a data type definition from the preprocessed source program;

a process for deciding, with reference to instantiation information in the data type definition table, whether the instance of a data type definition corresponding to the detected instantiation request of the data type definition has been generated or not; and

a process for generating the instance of the data type definition when it is decided that the instance of the data type definition does not have been generated, registering information representing the generation of the instance into the data type definition table as the instantiation information, and

suppressing generation of the instance of the data type definition when it is decided that the instance of the data type definition has been generated.

19. A program product for causing a computer to execute program language processing for translating source programs to generate an object program, comprising:

a process for executing preprocessing of source program input in translation units;

a process for scanning all the preprocessed source programs to be used as a source for generating the object program;

Sub A16

a process for deleting a duplicate data type definition from the source program with reference to a data type definition table arranged for one object program so as to suppress instantiation of a data type definition which has been instantiated as needed to optimize the source program, the table registering a data type definition for data or a function in the source program; and

a process for compiling the optimized source program in units of translation.

20. The program product according to claim 19, wherein the optimizing process includes:

a process for detecting a predetermined data type definition from the preprocessed source program;

a process for deciding whether definition information of the detected data type definition is registered into the data type definition table or not;

a process for registering the definition information of the data type definition into the data type definition table when it is decided that the definition information of the data type definition does not have been registered; and

a process for deleting the data type definition detected by the data type definition detection process from the preprocessed source program when it is decided that the definition information of the data type definition has been registered.

Add A17